INFORMATION AND CONTROL 18, 237–252 (1971)

# The Utility of Independent Subgoals in Theorem Proving

GEORGE W. ERNST

*Department of Computing and Information Sciences,*
*Case Western Reserve University, Cleveland, Ohio 44106*

This paper views theorem proving as a two-level process. The top level breaks theorems into subgoals which are achieved by contemporary theorem proving methods. A method for generating an AND/OR tree of subgoals is described. It is emphasized that all of the subgoals generated by this method are "independent." This is important because solving independent subgoals may yield an exponential savings. Examples are given to illustrate the subgoal generation method and its utility.

Since 1965, most efforts in mechanical theorem proving have been similar to those introduced by Robinson (1965) in that they use some variant of resolution (e.g., Slagle, 1967) and deal only with formulae in conjunctive normal form (CNF). This paper shows how goal-directed search, a technique used widely in problem solving (as opposed to theorem proving) programs, can be benefically incorporated into a theorem prover. The resulting structure is quite different from contemporary theorem provers although it does make use of well-known ideas in mathematical logic, such as miniscope (Wang, 1960) and natural deduction systems (Fitch, 1952).

Our theorem prover has a two-level structure. The top level breaks the theorem to be proven into subgoals. This process is discussed below in detail. The bottom level achieves the individual subgoals. We assume that conventional theorem-proving techniques are used for this purpose and do not, therefore, discuss them in this paper.

This paper assumes that the inputs to our theorem prover are a list of axioms, a list of previously proven theorems, a list of definitions and a theorem to be proven.[1] The definitions and the previously proven theorems are used at the top level to generate subgoals of the theorem to be proven. However, the axioms are used only in achieving individual subgoals.

[1] We distinguish between these three types of statements because they may have different syntactic forms and because they are used in different ways.

The first step in proving a theorem is to reduce it to normal form as described in the Section 1. Section 2 describes the subgoal generation method while the remaining sections discuss various aspects of it.

## 1. Simplified Miniscope

Before we attempt to prove a theorem, it is reduced to simplified miniscope normal form (SMNF). A formula is in miniscope (Wang, 1960) whenever the scopes of all of its quantifiers areas are as small as possible.[2] $\equiv$ is removed because quantifiers cannot be moved across $\equiv$. In addition to being in miniscope, a formula in SMNF has an atomic formula[3] as the argument of every occurrence of $\sim$. That is, it has no subexpressions of the form $\sim(A \,\&\, B)$, $\sim(A \vee B)$, $\sim(A \supset B)$.

A formula can be converted to SMNF by repeated application of the productions in Table I. In these productions, $\alpha$ denotes an individual variable. $A$ and $B$ denote arbitrary formulae except for the productions marked with $a^c$. In these productions, $B$ is an arbitrary formula containing no free occurrences of $\alpha$. $Q_e$ and $Q_f$ denote arbitrary strings of existential quantifiers and universal quantifiers, respectively. The expression to the left of a $\Rightarrow$ gives the form of a subexpression that can be replaced by a subexpression of the form given by the right of $\Rightarrow$. A formula to which none of these productions is applicable is in SMNF.

In applying productions 11 and 12, it may be necessary to commute and associate subexpressions. For example, $\forall x\, \forall y(P(x) \,\&\, (Q(y) \,\&\, R(x)))$ can be rewritten as $\forall y(\forall x(P(x) \,\&\, R(x)) \,\&\, Q(y))$ by one application of production 11.

We want to keep occurrences of $\supset$ in the formula if possible (for reasons which will becomes clear). Therefore, we will only use rule 5 whenever necessary. This is the reason for the restriction that production 5 can not be applied to a subexpression of the argument of a $\sim$. When the restriction is not fulfilled the formula is of the form

$$(\cdots \sim (\cdots \sim (A \supset B) \cdots) \cdots).$$

In this case, the first $\sim$ might cancel the second $\sim$ removing the necessity of applying production 5.

[2] Wang's miniscope is different from our SMNF because he uses the distributive laws for & and $\vee$ which sometimes further reduce the scope of quantifiers.

[3] An atomic formula is a predicate letter with its arguments each of which is a well-formed term involving function letters and individual variables.

TABLE I

A Set of Productions for Converting a Formula to SMNF[a]

| | |
|---|---|
| 1. | $A \equiv B \Rightarrow (A \supset B) \,\&\, (B \supset A)$ |
| 2. | $\sim\bar{A} \Rightarrow A$ |
| 3. | $\sim(A \,\&\, B) \Rightarrow \bar{A} \vee \bar{B}$ |
| 4. | $\sim(A \vee B) \Rightarrow \bar{A} \,\&\, \bar{B}$ |
| 5. | $\sim(A \supset B) \Rightarrow A \,\&\, \bar{B}$[b] |
| 6. | $\sim\forall\alpha A \Rightarrow \exists\alpha\bar{A}$ |
| 7. | $\sim\exists\alpha A \Rightarrow \forall\alpha\bar{A}$ |
| 8. | $\forall\alpha(A \,\&\, B) \Rightarrow \forall\alpha A \,\&\, \forall\alpha B$ |
| 9. | $\exists\alpha(A \vee B) \Rightarrow \exists\alpha A \vee \exists\alpha B$ |
| 10. | $\exists\alpha(A \supset B) \Rightarrow \forall\alpha A \supset \exists\alpha B$ |
| 11. | $\forall\alpha Q_f(A \vee B) \Rightarrow Q_f(\forall\alpha A \vee B)^{c,d}$ |
| 12. | $\exists\alpha Q_e(A \,\&\, B) \Rightarrow Q_e(\exists\alpha A \,\&\, B)^{c,d}$ |
| 13. | $\forall\alpha Q_f(A \supset B) \Rightarrow Q_f(\exists\alpha A \supset B)^{c}$ |
| 14. | $\forall\alpha Q_f(B \supset A) \Rightarrow Q_f(B \supset \forall\alpha A)^{c}$ |
| 15. | $\forall\alpha B \Rightarrow B^{c}$ |
| 16. | $\exists\alpha B \Rightarrow B^{c}$ |

[a] $Q_e$ denotes an arbitrary string of existential quantifiers and $Q_f$ denotes a string of universal quantifiers.

[b] This production is only applied to subexpressions that do not occur in the argument of a $\sim$.

[c] In these productions $B$ contains no free occurrences of $\alpha$.

[d] In these productions it may be necessary to commute or associate subexpressions.

Figure 1 gives an example of using the productions in Table I to convert a formula to SMNF.

1. $\forall xy(Q(y) \,\&\, R(x) \equiv P(x))$
2. $\forall xy(Q(y) \,\&\, R(x) \supset P(x)) \,\&\, \forall xy(P(x) \supset Q(y) \,\&\, R(x))$
3. $\forall x(\exists yQ(y) \,\&\, R(x) \supset P(x)) \,\&\, \forall x(P(x) \supset \forall yQ(y) \,\&\, R(x))$

FIG. 1. An example of using the productions of Table I to convert 1 to its SMNF, 3.

## 2. SUBGOAL GENERATION

After the theorem to be proven is in SMNF it is broken up into parts, called *subgoals*, in such a way that a proof of all of the parts constitutes a proof of the theorem. This subgoal generation technique is recursive since each

subgoal may in turn be broken up into further subgoals. Since there may be several different ways to break a subgoal into subgoals, the subgoal structure turns out to be an AND/OR tree which has been widely used in other problem solving programs. See Slagle (1970) for a discussion of this approach.

In subsection 2.1 we describe how the theorem to be proven can be broken into subgoals. The next two subsections discuss the use of definitions and previous proven theorems in subgoal generation.

### 2.1 *Subgoals Generated from the Theorem*

Suppose that $T$ is the SMNF of the theorem to be proven and that $T$ is of the form $Q(A \supset B)$, where $Q$ is a string of quantifiers (possibly null) and $A$ and $B$ are arbitrary formulae (which of course must be in SMNF). We allow $A$ to be an empty formula to represent the case where $T$ is of the form $Q(B)$ and where the main connective of $B$ is not $\supset$. If $Q$ contains an existential quantifier, then there are no subproblems, and we are stuck with proving $T$ by conventional methods. However, Section 5 discusses how subgoal generation might be extended to include special cases of an existential quantifier in $Q$.

If $Q$ consists entirely of universally quantified variables, there are three cases (depending on the form of $B$) in which $T$ can be broken up into subgoals:

*$B$ is of the form $B_1 \& B_2$.* In this case $T$ is broken up into the two subgoals $Q(A \supset B_1)$ and $Q(A \supset B_2)$.

*$B$ is of the form $Q_1(B_1 \supset B_2)$.* If $Q_1$ consists entirely of universally quantified variables, then $T$ is reduced to the single subgoal $QQ_1(A \& B_1 \supset B_2)$. The result is the same if $Q_1$ is empty. That is, if $B$ is of the form $B_1 \supset B_2$, the subgoal is $Q(A \& B_1 \supset B_2)$.

*$B$ is of the form $Q_1(B_1 \vee B_2)$.* If $Q_1$ consists entirely of universally quantified variables (possibly 0), then $T$ is equivalent either to the subgoal $QQ_1(A \& \sim B_1 \supset B_2)$ or to the subgoal $QQ_1(A \& \sim B_2 \supset B_1)$.[4]

An example will help to clarify our subgoal generation method, although in this subsection we only give an artificial example. Sections 2.2 and 2.3 have examples taken from group theory.

Suppose we want to derive

$$\forall uvw\ \exists x\ \forall z\ \exists y[R(u, w, v) \supset (P(x, u, w) \vee P(y, x, u) \vee \exists w(Q(z, u) \& Q(w, v) \& Q(v, u)))]$$

[4] These formulae may not be in SMNF. If not they are converted to SMNF before further consideration.

from some set of axioms. The SMNF of this formula is

$$\forall uvw[R(u, w, v) \supset (\exists x P(x, u, w) \vee \exists xy P(y, x, u) \vee (\forall z Q(z, u) \& \exists w Q(w, v) \& Q(v, u)))].$$

This can be broken into the following three subgoals:

$$\forall uvw[R(u, w, v) \& \forall x \bar{P}(x, u, w) \& \forall xy \bar{P}(y, x, u) \supset \forall z Q(z, u)],$$

$$\forall uvw[R(u, w, v) \& \forall x \bar{P}(x, u, w) \& \forall xy \bar{P}(y, x, u) \supset \exists w Q(w, v)],$$

$$\forall uvw[R(u, w, v) \& \forall x \bar{P}(x, u, w) \& \forall xy \bar{P}(y, x, u) \supset Q(v, u)].$$

This example illustrates that the success of our subgoal generation method depends on the extend to which existential quantifiers can be moved to the left of the main connective or inward toward the "literals" of the formula.[5] This does not illustrate the "OR" part of the AND/OR subgoal tree. However this will be clarified later, particularly in Section 2.3.

### 2.2 *Subgoals Generated from Definitions*

A definition is a statement of the form $P(t_1, t_2, \ldots, t_n) \equiv A$, where $P$ is a predicate letter and $t_1, t_2, \ldots, t_n$ are terms[6] containing individual variables that occur free in $A$, an arbitrary expression. A new subgoal can be generated from an old subgoal containing $P(t_1', t_2', \ldots, t_n')$ by replacing it with $A\sigma$, where $\sigma$ is the substitution such that $t_i\sigma = t_i'$ for $1 \leqslant i \leqslant n$. Of course, $\sigma$ need not exist. The new subgoal, itself may then be broken into other subgoals. An example will illustrate the utility of this type of subgoal generation.

Suppose we want to prove that the intersection of two subgroups is a subgroup. This involves the definition of a subgroup

$$S(x) \equiv \forall uv(u \in x \& v \in x \supset uv \in x) \& \forall u(u \in x \supset u^{-1} \in x)^7 \tag{1}$$

and the definition of intersection

$$u \in x \cap y \equiv u \in x \& u \in y. \tag{2}$$

[5] A literal is an atomic formula or its negation.

[6] Each term is a well formed expression involving function letters and individual variables.

[7] We consider these to be formulae in a first logic order. For example, $u^{-1} \in x$ is an abreviation for $e(i(u), x)$ where $e$ is a binary predicate and $i$ is a unary function.

The theorem to be proven, then, is

$$\forall xy(S(x) \,\&\, S(y) \supset S(x \cap y)) \tag{3}$$

which is already in SMNF.

By using the definition of subgroup, we generate a new subgoal which decomposes into two subgoals:

$$\forall xyuv(S(x) \,\&\, S(y) \,\&\, u \in x \cap y \,\&\, v \in x \cap y \supset uv \in x \cap y) \tag{4}$$

and

$$\forall xyu(S(x) \,\&\, S(y) \,\&\, u \in x \cap y \supset u^{-1} \in x \cap y). \tag{5}$$

The latter itself decomposes into two subgoals:

$$\forall xyu(S(x) \,\&\, S(y) \,\&\, u \in x \cap y \supset u^{-1} \in x) \tag{6}$$

and

$$\forall xyu(S(x) \,\&\, S(y) \,\&\, u \in x \cap y \supset u^{-1} \in y). \tag{7}$$

After replacing $S(y)$ and $u \in x \cap y$ by their definitions, (7) reduces to the subgoal

$$\begin{aligned} &\forall xyu(S(x) \,\&\, \forall uv(u \in y \,\&\, v \in y \supset uv \in y) \,\& \\ &\quad \forall u(u \in y \supset u^{-1} \in y) \,\&\, u \in x \,\&\, u \in y \supset u^{-1} \in y). \end{aligned} \tag{8}$$

This subgoal is solved in two steps by a theorem prover using binary resolution: (6) reduces to a subgoal that too gets solved in two steps. (4) decomposes into two subgoals each of whose proofs are three steps long.

The point of this example is that although there is considerable labor and decisions[8] at the subgoal generation level, the bottom-most subgoals are easily solved by conventional theorem-proving techniques.

### 2.3 *Subgoals Generated from Previously Proven Theorems*

The basic method in using previously proven theorems[9] to generate subgoals is given by the following case: Suppose that the theorem $T$ to be

[8] The types of decision that must be made are whether to substitute definitions for $S(x)$ or $S(y)$ or both or neither in (7). We consider these to be different subgoals because a theorem prover is faced with different decisions.

[9] This method can be used with axioms as well as previously proven theorems, but we have not found subgoals generated from axioms to be useful.

proven is of the form $Q_1(A_1 \supset Q_2 B_1)$ and there is a previously proven theorem of the form $Q_3(A_2 \supset Q_4 B_2)$, where $Q_1, Q_2, Q_3, Q_4$, contain no existential quantifiers. Then, if there is a substitution $\sigma$ such that $B_2\sigma = B_1$, the subgoal $Q_1(A_1 \supset Q_2(A_2\sigma))$ implies $T$.[10] An example will clarify this basic idea and then we will state the method in a more useful form.

Suppose we wanted to prove that the intersection of normal subgroups is normal, i.e.,

$$\forall xy(N(x) \,\&\, N(y) \supset N(x \cap y)). \tag{9}$$

We need the definitions (1) and (2) as well as the definition of normal

$$N(x) \equiv S(x) \,\&\, \forall uv(u \in x \supset (v^{-1}u)v \in x). \tag{10}$$

If we use the definition of normal, (9) reduces to the two subgoals

$$\forall xy(N(x) \,\&\, N(y) \supset S(x \cap y)) \tag{11}$$

and

$$\forall xyvu(N(x) \,\&\, N(y) \,\&\, u \in x \cap y \supset (v^{-1}u)v \in x \cap y). \tag{12}$$

By using the previously proven theorem (3), we reduce (11) to two subgoals

$$\forall xy(N(x) \,\&\, N(y) \supset S(x)) \tag{13}$$

and

$$\forall xy(N(x) \,\&\, N(y) \supset S(y)). \tag{14}$$

The latter is trivial after we substitute the definition of $N(y)$. That is,

$$\forall xy(N(x) \,\&\, S(y) \,\&\, \forall uv(u \in y \supset (v^{-1}u)v \in y) \supset S(y)) \tag{14}$$

is obviously a theorem. The proof of (13) is similar to that of (14) while the proof of (12) is much the same as that in subsection 2.2.

An obvious problem with this method of subgoal generation is when previously proven theorems are of the form $Q(A \supset B)$ and $B$ is not a single literal. Then the chances of matching $B$ to the consequent[11] of the theorem to be proven is very small. However, this problem can be avoided by writing a previously proven theorem as the conjunction of several parts. For example,

[10] This method is analogous to "backward chaining" in LT (Newell, Shaw, and Simon, 1963).

[11] If a statement is of the form $Q(A \supset B)$, where $Q$ is a string of quantifiers, then we call $A$ the antecedent and $B$ the consequent.

suppose the previously proven theorem is of the form $Q(A \supset B_1 \,\&\, B_2)$. Then, the theorem can be written as two separate theorems $Q(A \supset B_1)$ and $Q(A \supset B_2)$ if $Q$ contains no existential quantifiers. This process of breaking theorems into parts is the same as the subgoal generation process of subsection 2.1. Of course, the use of the parts is entirely different. To prove a theorem consisting of the conjunction of parts, one needs to prove each part while any one part of a previously proven theorem is valid; thus only one part needs to be used to generate subgoals. For this reason, the use of previously proven theorems (or parts thereof) is the main source of "OR-nodes" in the AND/OR subgoal tree.

## 3. Utility of Independent Subgoals

The last section describes a method for subdividing a theorem into subgoals. It is quite obvious that this method is logically sound since it is based upon well-known equivalences, e.g., $A \supset B \,\&\, C \equiv ((A \supset B) \,\&\, (A \supset C))$. However, much less obvious is the reason why we focus on a particular class of subgoals; certainly there are many other ways to break a theorem into subgoals.

The most important property of the subgoals produced by the method described in the last section is that they are "independent." We say that the subgoals are *independent* when they have no variables in common. This is important because independent subgoals can be proven independently. For example, $\forall x P(x) \supset \forall x(Q(x) \,\&\, R(x))$ can be proven by finding contradictions in both $\forall x P(x) \,\&\, \bar{Q}(a)$ and $\forall x P(x) \,\&\, \bar{R}(a)$.[12] These two subgoals are independent. However, $\forall x P(x) \supset \exists x(Q(x) \,\&\, R(x))$ can be proven by finding contradictions in both $\forall x P(x) \,\&\, \forall y \bar{Q}(y)$ and $\forall x P(x) \,\&\, \forall y \bar{R}(y)$ *provided that* the same value for $y$ is substituted in both.[13] That is, these two subgoals have $y$ in common, and therefore are not independent.

Our subgoal generation method attempts to subdivide a theorem into the conjection of independent subgoals. This produces a possible exponential saving! That is, suppose that a theorem breaks up into $p$ independent sub-

[12] These subgoals have been formulated in the format of contemporary theorem provers. That is, we have negated the subgoals and put in the Skolem function $a$.

[13] Both of these subgoals are of the form "Find an $\alpha$ such that $A$." For example, the former is of the form (before negation) "find a $y$ such that $\forall x P(x) \supset Q(y)$." Green (1969) gives a method for finding such a $y$ by conventional theorem proving techniques. Thus, if one finds $t$ to be such a $y$, one merely has to solve $\forall x P(x) \supset R(t)$. The problem is that there may be many different values of $y$. Thus one must try many alternate solutions to the first subgoal in order to find a $y$ which can be used in the solution of the second subgoal.

goals, each of whose solution is of length $n$.[14] Then if $k$ is the average number of possibilities that must be examined at each stage of the proof, the total number of inferences made in exhaustively searching for a proof is approximately $pk^n$. If the theorem were not broken up into independent subgoals, the total number of inferences that must be examined would be about $k^{pn}$.

This is the best case analysis because it assumes that the length of the proof of the theorem is equal to the sum of the lengths of the proofs of subgoals of which all are the same length. In general, this is not true. The proofs of some subgoals are longer than the proofs of others and there may be duplication in the proofs of the subgoals. In fact, there is no guarantee that the proof of a subgoal is shorter than the proof of the theorem taken as a whole. It may even be the case (as discussed below) that a subgoal is not valid when the theorem to be proven is. However, in many cases there is an exponential saving.

Note that we can afford to look at many subgoals and still achieve an exponential saving. Suppose there were $k^n$ different possible subgoals. Then using an exhaustive search, we only need to look at $k^n\ k^n = k^{2n}$ different subgoal solutions, where $k$, $n$ and $p$ are defined, as above. Thus, if $p > 2$, we still have an exponential saving. Of course, an exhaustive search of an AND/OR tree is ridiculous because failure at a branch from an AND node eliminates the other branches from the node.

An important aspect of our method is that the subgoal tree is relatively small. Conceivably, the subgoal tree could be large enough to use up all of the savings due to our breaking the theorem into parts. This is not the case in any of the examples that we have looked at. There are two reasons why subgoal generation is quite efficient: with the exclusion of those subgoals generated from previously proven theorems, the subgoal tree is finite. The number of subgoals generated from the theorem to be proven is certainly less than the number of literals that it contains. Substitution for definitions is finite if the definitions are not recursive.

Another good feature of subgoal generation is that inferences are made in generating subgoals. This reduces the inferences that must be made in achieving the individual subgoals that constitute a proof. Below we give an example that yields an exponential saving; the appendix contains more examples to give the reader a better idea of the efficiency and generality of our method. Of course, in many cases our method is only trivially applicable in that the subgoal tree consists of a single subgoal of the theorem to be proven.

[14] From this analysis, our subgoal generation method is very similar to what Minsky (1963) calls planning.

In Section 2 we showed how (3) can be divided into four subgoals. Two of these can be proven with two binary resolutions while the others require three. Thus, each subgoal can be solved by an exhaustive search of a tree of depth three, i.e., the total number of binary resolutions that needs to be examined is $4k^3$, where $k$ is the average number of possible binary resolutions at each point in the proof.

Since the subgoal tree has 76 terminal nodes, an exhaustive search only needs to look at $76k^3$. Since $k$ is approximately ten, exhaustive search must look at $10^5$ binary resolutions.

Using the formulation in Fig. 3b the length of the proof is 17 and $10^{17}$ binary resolutions must be examined.[15] (Actually, for this formulation $k$ is considerably larger than ten because it contains clauses that constitute the definitions of subgroup and intersection.) Thus, we have a saving of a factor of 3 in the exponent.

## 4. *Natural Form*

Our subgoal generation method attempts to keep statements as close to their original form as possible. This is the reason that $\supset$ is not replaced by its definition in terms of $\vee$ and & except under special circumstances. The rationale is that people use $\supset$ in such a way as to simplify statements and theorem provers should make use of such information whenever possible. This is merely a heuristic (which is also used by natural deduction systems) that may prove to be useful, and the usefullness of this heuristic is certainly an open question. A brief statement in its support follows.

Often theorems are of the form $Q(A \supset B)$ where both $A$ and $B$ are in CNF. However, to convert the entire theorem to CNF is a mess because the theorem is of the form $Q(\bar{A} \vee B)$ where $\bar{A}$ is in disjunctive normal form. Suppose, for example, we substitute the definition of $S$ into (3) wherever possible. In the result, $A$ consists of 4 conjuncts and $B$ consists of 2 conjuncts, and the theorem's CNF is complicated.

Converting the negation of such a theorem to CNF (which must be done for contemporary theorem provers) is almost as tragic. That is the negation of $A \supset B$ is $A \,\&\, \bar{B}$ and $\bar{B}$ is in disjunctive normal form because $B$ is in CNF. Figure 2 gives the CNF of $\sim$(3) to illustrate how much more complicated it is than (3). Figure 3a gives the CNF of $\bar{S}(x \cap y)$ when the Skolem functions

[15] The sum of the lengths of the subgoals is $3 + 3 + 2 + 2 < 17$, even though 17 is the shortest proof we could find. The missing 7 binary resolutions were done in generating the subgoals. Thus, some of the effort in generating subgoals is paid for because the total number of inferences needed to prove the theorem is smaller.

$$\left.\begin{array}{l}\{u \notin a, v \notin a, uv \in a\}\\ \{u \notin a, u^{-1} \in a\}\end{array}\right\} S(a)$$

$$\left.\begin{array}{l}\{u \notin b, v \notin b, uv \in b\}\\ \{u \notin b, u^{-1} \in b\}\end{array}\right\} S(b)$$

$$\left.\begin{array}{l}\{d \in a \cap b, c \in a \cap b\}\\ \{f \in a \cap b, c \in a \cap b\}\\ \{df \notin a \cap b, c \in a \cap b\}\\ \{d \in a \cap b, c^{-1} \notin a \cap b\}\\ \{f \in a \cap b, c^{-1} \notin a \cap b\}\\ \{df \notin a \cap b, c^{-1} \notin a \cap b\}\end{array}\right\} \bar{S}(a \cap b)$$

$$\left.\begin{array}{l}\{u \notin x, y \notin y, u \in x \cap y\}\\ \{u \notin x \cap y, u \in x\}\\ \{u \notin x \cap y, u \in y\}\end{array}\right\} \text{definition of } \cap$$

FIG. 2. The CNF of $\sim\forall xy(S(x) \,\&\, S(y) \supset S(x \cap y))$.

(a) $\{c \in a \cap b\}$
$\{d \in a \cap b, c \in a \cap b\}$
$\{cd \notin a \cap b, c \in a \cap b\}$
$\{c \in a \cap b, c^{-1} \notin a \cap b\}$
$\{d \in a \cap b, c^{-1} \notin a \cap b\}$
$\{cd \notin a \cap b, c^{-1} \notin a \cap b\}$

(b) $\{c \in a \cap b\}$
$\{d \in a \cap b, c^{-1} \notin a \cap b\}$
$\{cd \notin a \cap b, c^{-1} \notin a \cap b\}$

FIG. 3. (a) The CNF of $\sim\forall uv[(u \in a \cap b \,\&\, v \in a \cap b \supset uv \in a \cap b) \,\&\, (u \in a \cap b \supset u^{-1} \in a \cap b)]$. (b) The result of applying the subsumption principle to (a).

are added only after converting the fomulae to prenex. The subsumption principle (Robinson, 1965) reduces this to Figure 3b which is not much worse than the definition of $S(x \cap y)$. However, Figs. 3a and 3b both obscure the fact that they are $\bar{S}(x \cap y)$.

## 5. *Completeness*

Completeness of our subgoal generation method is not really an issue. The methods in subsections 2.1 and 2.2 are inherently complete because the conjunction of the subgoals are equivalent to the theorem to be proven.[16] Thus this part of the subgoal generator can be viewed as a simplification of theorems before we attempt to prove them.

[16] We are assuming that the subgoal generator considers substituting for all definitions because definitions are not given to the theorem prover that proves an individual subgoal.

That part of the method developed in subsection 2.3 is inherently incomplete. That is, if the theorem to be proven is $A_1 \supset B$ and a previously proven theorem is $A_2 \supset B$, then it is certainly not necessary for $A_1 \supset A_2$ to be valid. However, if it were valid, then so would $A_1 \supset B$ be. This merely indicates that we must attempt subgoals not involving previously proven theorems as well as those which do (unless the previously proven theorem is of the form $A_2 \equiv B$).

6. *Extension of the Method*

In Section 3 we discussed the importance of independent subgoals and showed why existential quantifiers yield dependent subgoals. Since we do not know how to process dependent subgoals efficiently we have left these to be processed by standard theorem proving techniques. Again, there are special cases that we do know how to handle.

The most interesting case involves "there exists a unique $x$ such that..." For example, suppose that we want to prove

$$\exists y(Q(y) \,\&\, \forall u(Q(u) \supset u = y)). \tag{15}$$

This can be proven by finding, contradictions in both

$$\forall y \bar{Q}(y) \tag{16}$$

and

$$\forall y(Q f(y) \,\&\, f(y) \neq y), \tag{17}$$

*provided that* the same value is used for $y$ in both contradictions. However, if (15) is valid, then there is only one $y$ which makes (16) valid. Thus, first we find a contradiction in (16). From this we get a value $t$ for $y$.[17] Then (15) is valid if and only if there is a contradiction in

$$Q(f(t)) \,\&\, f(t) \neq t,$$

which is the result of substituting $t$ for $y$ in (17). The point is that although (16) and (17) are dependent they can be solved independently because if (15) is valid we know that $t$ is the correct value for $y$ and never need to search for alternative solutions to (16). Thus, this case also gives us an exponential saving (the exponent is halved) over standard theorem provers.

Of course, once we have proven (15), we can use it to remove the existential quantifiers from other theorems thus making them more susceptible to our subgoal generation method. For example, $\exists y(Q(y) \,\&\, R(y))$ is equivalent to $R(t)$ once we have proven (15).

[17] Green (1969) gives a method for finding such a $t$.

There are other cases where there is a saving in processing dependent subgoals. For example, suppose we wanted to prove $\exists x(P(x) \& Q(x))$. If we found in attempting to prove $\exists x P(x)$ that $\forall x P(x)$ is valid, then we could approach $\exists x Q(x)$ independently. However, such cases that we know of are *ad hoc* and/or unlikely.

## 7. *Conclusion*

The main point of this paper is that when a theorem can be broken into independent subgoals there is an exponential saving in the search for a solution. Dependent subgoals arise from existential quantifiers governing variables in several literals in the consequent of the theorem to be proven. Thus, the success of our subgoal method depends upon the extent to which existential quantifiers can be moved in front of literals or into the antecedent of the theorem. The scope of quantifiers is minimized by reducing the theorem to SMNF and the subgoal generation procedure puts existential quantifiers into the antecedent whenever possible.

The quantifier "there exists a unique" can be handled nicely by our method. If the consequent of the theorem contains this quantifier, the theorem can be broken into two subgoals yielding an exponential savings. If a previously proven theorem contains this quantifier, it can be used to remove existential quantifiers in formulae; thus making some theorems more susceptible to our subgoal method.

A possibly valuable side effect of our subgoal method is that it does not convert formulae to disjunctive or conjunctive normal form. Such conversions are dangerous because often they result in a much more complicated formula.

Our subgoal method is complete. However, some of the subgoals that can be generated are not valid and care must be taken not to spend too much effort on these subgoals.

This paper views theorem proving as a two-level process. The top level searches an AND/OR subgoal tree while the bottom level proves subgoals by conventional theorem proving techniques. These two levels should not operate independently but should get "feedback" from one another. This paper does not discuss this issue, but the literature on searching AND/OR trees is certainly relevant. Along these lines, it appears to us that as many decisions as possible should be made at the subgoal generation level. For example, it appears beneficial to decide at the top level which definitions and previously proven theorems can be used in attempting a subgoal. However, more research is needed on such issues.

## Appendix

This appendix contains two examples of our subgoal generation method.

Example 1. In a group the equations $xa = b$ and $ay = b$ have unique solutions.

This theorem can be expressed in first-oder logic as

$$\forall ab\, \exists xy(xa = b \,\&\, ay = b \,\&\, \forall z(za = b \supset z = x) \,\&\, \forall z(az = b \supset z = y)).$$

The SMNF of this statement is

$$\forall ab\, \exists x(xa = b \,\&\, \forall z(za = b \supset z = x)) \,\&\, \forall ab\, \exists y(ay = b \,\&\, \forall z(az = b \supset z = x)).$$

This statement breaks up into two pairs of subgoals; each pair is sequential as described in Section 6.
Let $A$ be the following axioms of a group:

$$\forall x(xe = x \,\&\, ex = x),$$
$$\&\, \forall x(x^{-1}x = e \,\&\, xx^{-1} = e),$$
$$\&\, \forall xyz(x(yz) = (xy)z).$$

Then the first pair of subgoals is

$$A \supset \forall ab\, \exists x(xa = b)$$

and

$$\forall abz(A \,\&\, za = b \supset z = x).$$

The solution to the former takes 4 steps[18] and produces $ba^{-1}$ for the value of $x$. Substituting this into the latter, it becomes

$$\forall abz(A \,\&\, za = b \supset z = ba^{-1})$$

which can be solved in 5 steps.

The other pair of subgoals are very similar and get solved in 4 and 5 steps, respectively.

Since the longest proof for a subgoal is 5 steps, a subgoal can be solved by an exhaustive search of $k^5$ where $k$ is the average number of inferences that can be made at a given point. In this problem, 7 subgoals can be generated.

[18] By step we mean a binary resolution or the substitution of an expression for its equal.

Therefore, $7 \cdot k^5$ is an upperbound on the number of inferences that need to be examined.

We do not know how long the shortest proof is when the theorem is taken as a whole. We conjecture that it is close to 18; i.e., we conjecture that our method introduces a savings of almost a factor of 4 in the exponent. The reason the proof is so long is that the CNF of the negation is a mess. That is, a contemporary theorem prover would attempt to find a contradiction in

$$\begin{aligned} A \,\&\, (xa \neq b \vee ay \neq b \vee f(x,y)\, a = b \vee ag(x,y) = b) \\ \&\, (xa \neq b \vee ay \neq b \vee f(x,y)\, a = b \vee g(x,y) \neq y) \\ \&\, (xa \neq b \vee ay \neq b \vee f(x,y) \neq x \vee ag(x,y) = b) \\ \&\, (xa \neq b \vee ay \neq b \vee f(x,y) \neq x \vee g(x,y) \neq y), \end{aligned}$$

where $f$ and $g$ are functions and $a$ and $b$ are constants.

EXAMPLE 2. A subgroup is normal iff for all $u$ its left coset with respect to $u$ equals its right coset with respect to $u$. The definitions of normal and subgroup are given in the paper; the definitions or right and left cosets, respectively, are

$$x \in rc(u, Y) \equiv \exists v(v \in Y \,\&\, x = vu)$$

and

$$x \in lc(u, Y) \equiv \exists v(v \in Y \,\&\, x = uv).$$

If we use these definitions, the theorem is

$$S(Y) \supset [N(Y) \equiv \forall u(rc(u, Y) = lc(u, Y))].$$

Assuming that $A$ is the axioms for a group (as defined above), we break this theorem into 4 subgoals:

1. $A \,\&\, S(Y) \,\&\, \forall x(x \in Y \supset \forall u(u^{-1}(xu) \in Y))$
   $\&\, \exists z(z \in Y \,\&\, v = zu) \supset \exists z(z \in Y \,\&\, v = uz).$
2. $A \,\&\, S(Y) \,\&\, \forall x(x \in Y \supset \forall u(u^{-1}(xu) \in Y))$
   $\&\, \exists z(z \in Y \,\&\, v = uz) \supset \exists z(z \in Y \,\&\, v = zu).$
3. $A \,\&\, S(Y) \,\&\, \forall u(rc(u, Y) = lc(u, Y)) \supset S(Y).$
4. $A \,\&\, S(Y) \,\&\, \forall uv(\exists z(z \in Y \,\&\, v = zu) \supset \exists z(z \in Y \,\&\, v = uz))$
   $\forall uv(v \in lc(u, Y) \supset v \in rc(u, Y)) \,\&\, x \in Y \supset u^{-1}(xu) \in Y.$

The subgoal generation is quite straightforward except in the use of the definition of *rc* of *lc*. The substitution of these definitions requires the use of $\forall x(x \in y \equiv x \in z) \equiv y = z$. That is, $lc(u, Y) = rc(u, Y)$ must be written as $\forall x(x \in lc(u, Y) \equiv x \in rc(u, Y))$ before we substitute for *lc* or *rc*.

Subgoals 1 and 2 take 7 steps each; 3 takes 1 step while 4 takes 10 steps. Less than 200 subgoals can be generated for this problem. Thus, our exhaustive search of $200k^{10}$ suffices for this theorem.

We conjecture that the shortest proof of the theorem taken as a whole is about 30. The reason, again, is that the CNF of the definitions and the negation of the theorem is a mess. The number of clauses that a theorem prover is given for the above 4 subgoals are 10, 10, 8, 11, respectively. On the other hand, the negation of the theorem (with necessary definitions) requires 21 clauses. This count excludes the definition of subgroup, which is five clauses long, because it is not used in the proof of the theorem. In addition, the average length of a clause is longer when the theorem is taken as a whole.

## Acknowledgments

This work was sponsored by U.S.A. Air Force Office of Scientific Research under grant AF-OSR-125-67 and by the National Science Foundation under grant GJ-1135, to which agencies the author offers his gratitude.

Received: May 22, 1970

## References

Fitch, F. B. (1952), "Symbolic Logic: An Introduction," Ronald Press, New York.

Green, C. (1969), Applications of theorem proving to problem solving, "Proc. Internat-Joint Conf. Artificial Intelligence," (Walker, D. E. and Norton, L. M., Eds.), Mitre Corp., Bedford, Mass.

Minsky, M. (1963), Steps toward artificial intelligence, "Computers and Thought," (Feigenbaum, E. A. and Feldman, J., Eds.), McGraw-Hill, New York.

Newell, A., Shaw, J. C., and Simon, H. A. (1963), Empirical explorations with the logic theory machine, "Computers and Thought," (Feigenbaum, E. A. and Feldman, J., Eds.), McGraw-Hill, New York.

Robinson, J. A. (1965), A machine-oriented logic based on the resolution principle, *J. ACM* **12**, 32–41.

Slagle, J. (1970), Heuristic search programs, "Theoretical Approaches to Non-Numerical Problem Solving," Springer-Verlag, New York.

Slagle, J. (1967), Automatic theorem proving with renamable and semantic resolution, *J. ACM* **14**, 687–697.

Wang, H. (1960), Toward mechanical mathematics, *IBM J. Res. Develop.* **4**, 224–268.